Wired for

Management

Intel® Desktop Management Interface (DMI) 2.0s Primer

intel®

Desktop Management Interface (DMI) 2.0s Primer

This paper, intended for customers and developers of Desktop Management Interface (DMI) products, provides an overview of DMI 2.0s and the answers to frequently-asked questions. We assume the reader is familiar with DMI.

The properties of a managed system can be exposed through DMI 2.0. DMI 2.0 does not, however, restrict which actions a DMI management application can perform. Currently, an unauthorized user can start a DMI management application from any computer on the network and perform DMI commands. With the growing number of DMI-based systems deployed in the marketplace, there has been a strong demand by vendors and users for a more secure version of DMI 2.0. In August 1997, the Desktop Management Task Force, Inc. (DMTF) created a Working Group to address this demand.

The DMI 2.0s Working Group is chartered with extending the DMI 2.0 Specification to define standard interfaces to secure the interaction between the DMI Service Provider, management applications and component instrumentation. The DMI 2.0s Working Group includes representatives from Auspex, Compaq, Dell, Hewlett-Packard, IBM, Intel, NCR and SCO. The Working Group created an extension to the DMI 2.0 Specification (http://www.dmtf.org/tech/specs.html), which was approved by the DMTF in April 1998.

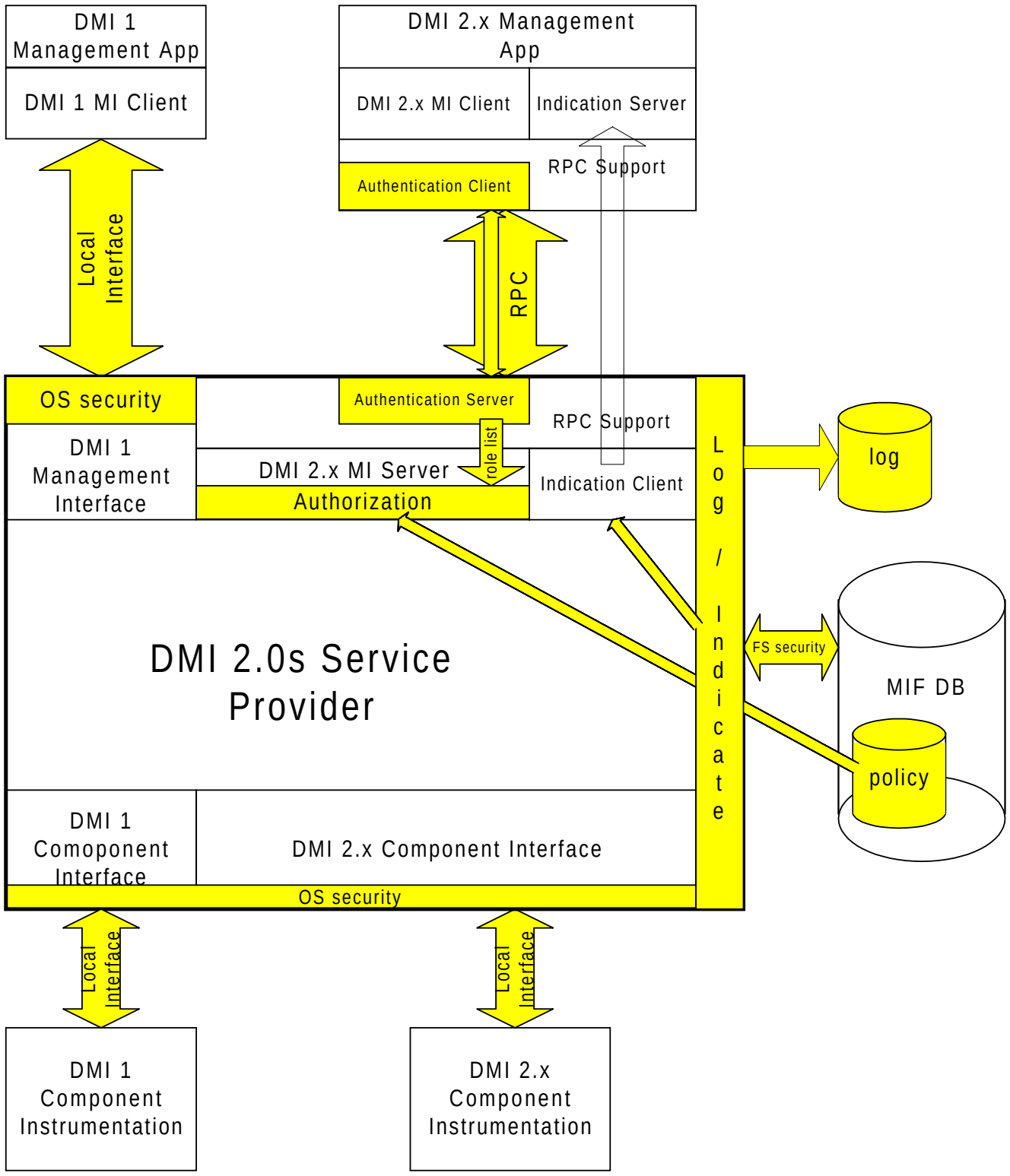
The DMI 2.0s Specification is an extension of the DMI 2.0 Specification and defines mechanisms to control local and remote access to the DMI interfaces. The remote access control mechanism is defined on top of standard Remote Procedure Call (RPC) mechanisms, whereas the local access control mechanism is defined on top of operating system mechanisms. DMI 2.0s does *not* specify a format for identities nor a cryptosystem to verify those identities, but rather relies on those provided by the RPC and the operating system.

The DMI 2.0 component and management interfaces are unchanged in the DMI 2.0s Specification. The functions and parameters of the Management Interface and the Component Interface in DMI 2.0s are identical to those of DMI 2.0; that is, the Interface Descriptive Language (IDL) of DMI 2.0s is identical to that of DMI 2.0. Therefore, DMI 2.0s Service Providers are compatible with existing DMI management applications and component instrumentation. DMI 2.0s modifies the behavior of the Service Provider by specifying an authorization mechanism based on the identity of the user accessing DMI and a policy. DMI 2.0s defines a flexible policy that controls which interactions between DMI entities are permitted. The policy is a DMI table stored in the Management Information Format (MIF) database, in which each row defines access rights. The policy can be modified through DMI commands, and the MIF database is protected through file system access mechanisms. In addition, the DMI 2.0s Service Provider may generate an indication or a log entry when certain operations are performed. The block diagram later in this document outlines the mechanisms used by DMI 2.0s to provide these features.

The security model of DMI 2.0s is *role*-based: users are granted privileges according to their role in the organization (system administrator, network manager, help desk, and so on). Authentication yields the list of roles of a user, which is used by the DMI 2.0s Service Provider for authorization. Upon registration, a DMI 2.0s management application proves the identity of its user to the DMI 2.0s Service Provider. Actually, the management application proves its identity to the RPC client stack, which securely transmits the identity to the RPC server stack, which in turn provides it to the DMI 2.0s Service Provider. Implementations of DMI 2.0s that are based on operating system authentication can use operating system user groups to associate users with roles.

To avoid defining yet another key or password authentication system, DMI 2.0s is based on existing authentication infrastructures. The most widely deployed authentication infrastructure is the operating system user login process. Therefore, DMI 2.0s implementations may use operating system primitives for authentication, although the DMI 2.0s Specification does not preclude using operating system-independent authentication systems, such as digital certificates. The RPC infrastructure can be configured to guarantee the *integrity* and *privacy* of RPC sessions. This way, no third party can intercept or tamper with the data exchanged. Management applications accessing the DMI 2.0s Management Interface through a non-authenticated RPC are assigned a default role. The policy defines the privileges of the default role, so that the privileges of non-authenticated users and legacy DMI 2.0 management applications can be configured.

When a management application attempts to perform a DMI command, the Service Provider looks up a *policy* to *authorize* the command. Authorization is a Boolean function of the command, its parameters, the roles of the user and the policy. Some commands, such as DmiRegister, are allowed for all management applications. The DMI 2.0s Service Provider has a configurable log feature that can log commands performed. This ensures that users managing a system are accountable for the commands they perform. Log entries are stored through a standard logging system, such as UNIX* syslog. The Service Provider may be configured to generate indications upon performing any command, such as an attempt to exceed privileges. Since component instrumentation controls the behavior of DMI components, it is one of the most powerful and vulnerable elements in the system. The DMI 2.0s Service Provider protects component instrumentation from invocation by unauthorized users through the Management Interface access-control mechanism. The Service Provider can also be configured to allow only operating system-privileged processes to register component instrumentation.


DMI 1
Management App
DMI 1 MI Client
DMI 2.x Management
App
DMI 2.x MI Client
Indication Server
RPC Support
Authentication Client
Local
Interface
RPC
OS security
Authentication Server
RPC Support
role list
DMI 1
Management
Interface
DMI 2.x MI Server
Indication Client
Authorization
Log / Indicate
log
FS security
MIF DB
policy
DMI 2.0s Service
Provider
DMI 1
Comoponent
Interface
DMI 2.x Component Interface
OS security
Local
Interface
Local
Interface
DMI 1
Component
Instrumentation
DMI 2.x
Component
Instrumentation

Questions and Answers

Q: What are the security features of DMI 2.0s?
A: DMI 2.0s defines the following security features:

- authentication of the user invoking the management application by the Service Provider
- authorization of management application operations by the Service Provider
- integrity and privacy of the RPC DMI management session
- logging of management application operations by the Service Provider

This means that the Service Provider may enable or disable management application commands based on the user identity, and keep track of the commands performed. DMI 2.0s also controls access by management applications running locally on the managed PC. The RPC stack may also provide integrity and privacy.

Q: What are you protected from? What you do leave unprotected?
A: DMI 2.0s protects management information from unauthorized access over the network. DmiRegister is allowed for any management application; indications can be received by any management application.

Q: Is DMI 2.0s a standard? How is it related to other industry standards?
A: DMI 2.0s is an industry standard sanctioned by the DMTF in April 1998. DMI 2.0s addresses protection of DMI information, and is independent of other standards such as Simple Network Management Protocol (SNMP) v.3 user level security.

Q: Do you use a public key scheme?
A: The authentication process used by DMI 2.0s is based on primitives provided through the RPC infrastructure, such as operating system login mechanisms. Different operating systems use different key schemes. Some authenticated RPCs use public key schemes for authentication.

Q: What are the Specification and implementation roadmaps?
A: The DMI 2.0s Specification was approved by the DMTF in April 1998. Intel is developing a reference implementation of the Service Provider for Microsoft Windows* operating systems. A Beta version of the Intel DMI 2.0s SDK is available at http://developer.intel.com/ial/wfm/tools/dmi20s

Q: What is the effect on component instrumentation developers?
A: None. DMI 2.0 and DMI 1.x instrumentation runs unchanged under the DMI 2.0s Service Provider.

Q: Are there any export limitations?
A: DMI 2.0s implementations are based on existing RPC security infrastructure, such as the operating system login primitives. DMI 2.0s implementations based on exportable RPC systems can be exported. Export limitations of RPC systems and operating systems are outside the scope of this Q&A.

Q: Is DMI 2.0s operating system-specific?
A: DMI 2.0s implementations can be based on RPCs that use operating system security primitives, such as login primitives, or on operating system-independent protocols, such as Kerberos or X.509 certificates.

Q: What about interoperability between clients and servers on different operating systems?
A: Interoperability is the ability for management applications and Service Providers residing on different operating systems to establish a remote DMI 2.0s session. Implementations of DMI 2.0s using the same authentication mechanism on different operating systems are interoperable. Authentication clients for several operating systems are available, such as the Novel NetWare* client for Windows. These clients can be used to implement DMI 2.0s clients that work across heterogeneous operating systems. Note that different types of RPC, such as Open Network Computing (ONC) and Distributed Computing Environment (DCE), are not interoperable.

Q: What about Windows NT* 5.0?
A: Windows NT 4.0 authentication is based on LAN Manager*. Windows NT 5.0 will introduce two additional authentication mechanisms: private key with Kerberos and public key with X.509 certificates. The first implementation of DMI 2.0s for Windows is based on LAN Manager, and therefore runs on Windows NT 4.0, Windows NT 5.0 and Windows 9x.

Q: How do you deploy DMI 2.0s systems? How complicated is it?
A: DMI 2.0s can be installed on new systems, or existing DMI 2.0 systems can be upgraded to DMI 2.0s. When upgrading a DMI 2.0 system to DMI 2.0s, the components installed in the MIF database are preserved. Upon installation, a default policy is installed, which can be modified. Upgrading a managed system from DMI 2.0 to DMI 2.0s involves replacing the Service

Provider. Upgrading a DMI 2.0 management application to DMI 2.0s involves replacing the RPC interface library or the client front end (CFE).

Q: What are the effects of DMI 2.0s on system administrators?
A: System administrators configure DMI 2.0s to control user access to certain DMI operations according to the role of that user. The role is a characteristic of a user, usually related to his or her actual role in the organization (for example, technician, application support, network manager, and so on). Users with the same role have the same privileges. In DMI 2.0s, the list of users with a certain role can be maintained using an operating system group. Operating system groups and their membership lists are maintained with standard operating system tools (User Manager on Windows NT, /etc/groups on UNIX, SYSCON on NetWare). The policy (the privileges of each role) is maintained as a table in the DMI 2.0s MIF database.

Q: How are mobile PCs supported?
A: There is no specific support for access to mobile PCs in DMI. The DMI 2.0s Service Provider is an RPC server that runs on the managed PC. The Service Provider is resilient to connection/disconnection from the network and to suspend/resume power saving operations. The DMI 2.0s Service Provider can work over Point-to-Point (PPP) connections. However, a PC that is disconnected from the network cannot send indications or be managed through DMI.

Q: What is the migration path from DMI 2.0 managed systems to DMI 2.0s?
A: DMI 2.0s affects both the managed PCs on which the DMI Service Provider runs and the management applications that remotely manage them. On the managed system, the DMI 2.0 Service Provider should be replaced with a DMI 2.0s Service Provider and a policy set. Both operations are performed by the DMI 2.0s Service Provider installation program. Instrumentation runs unchanged.

Q: What is the management applications migration path? How difficult is it?
A: DMI 2.0 management applications can be used unchanged with DMI 2.0s Service Providers, and will be granted default privileges. If the DMI 2.0 Client Front End library is replaced with the DMI 2.0s version, DMI 2.0 management applications will be granted privileges according to the user that starts the management application, and the management session may enable integrity and privacy.

Q: What is the compatibility with DMI 2.0 Service Provider/Instrumentation?
A: A management application configured to establish a secure session with the DMI 2.0s Service Provider works correctly with a DMI 2.0 Service Provider. Instrumentation runs unchanged under both DMI 2.0 and DMI 2.0s Service Providers.

Q: How is DMI v1.x supported?
A: Support of the DMI 1.x Management Interface is optional in DMI 2.0s. The DMI 1.x Component Interface is supported, so instrumentation written for DMI 1.x runs under DMI 2.0s.

Q: Is DMI 2.0s compatible with other DMI security mechanisms?
A: Several products have implemented proprietary security mechanisms to control access to DMI. DMI 2.0s does not attempt to maintain compatibility with these mechanisms.

Q: Does the DMI 2.0s Specification enforce interoperability?
A: The DMI 2.0s Specification defines the external interfaces to the DMI 2.0s Service Provider in terms of RPCs, and requires that the RPC used support authentication. Implementations of DMI 2.0s using the same authentication mechanism on different operating systems are interoperable.

Terminology

- *Authentication:* The process of reliably verifying the identity of a communicating party. For example, a login process is an Authentication of a user by an operating system. The process by which one communication endpoint verifies that the other party is indeed who he or she claims to be.
- *Authorization:* The process by which a provider decides whether to honor a request (usually according to the authenticated identity of the requesting party and a policy). For example, a file system may check the permission list associated with each file before authorizing a user to access a file. This permission list maps between file operations (like read or write) and user groups.
- *Privacy:* A property of a communication protocol that ensures that the data exchanged can be disclosed only by its intended recipient. That is, the data will remain opaque to any unauthorized party trying to decode it.
- *Integrity:* A property of a communication protocol that ensures that data received wasn't modified by an unauthorized principal and is identical to the data that was transmitted. Integrity mechanisms can be based on a checksum computed on the transmitted message. Messages received with an incorrect checksum are discarded.

Author: Marc Jalfon - Intel Corporation

717044-001